9. What are the different vulnerability types, and how do you assess their severity?

🛠️ Types of Vulnerabilities & Assessing Their Severity

Understanding different types of vulnerabilities and how to assess their severity is **crucial** in cybersecurity. This guide covers **common vulnerability types**, their **risk assessment methods**, and how to prioritize them effectively. 🚀

🔍 1. Common Types of Vulnerabilities

Vulnerabilities fall into different categories based on their nature and impact. Below are some key types:

A. Software Vulnerabilities

These exist due to flaws in **code, logic, or design**.
✅ **Examples:**

- **Buffer Overflow** – Excessive data overwriting memory
- **SQL Injection (SQLi)** – Injecting malicious SQL queries
- **Cross-Site Scripting (XSS)** – Injecting JavaScript into web pages
- **Remote Code Execution (RCE)** – Running attacker-controlled code

B. Network Vulnerabilities

These involve weaknesses in network configurations or protocols.
✅ **Examples:**

- **Open Ports & Misconfigurations** – Exposing unnecessary services
- **Unencrypted Traffic** – Lack of SSL/TLS
- **Weak Authentication** – Default or weak credentials

C. Hardware & Firmware Vulnerabilities

Weaknesses in **physical devices or embedded software**.
✅ **Examples:**

- **Side-Channel Attacks** – Exploiting hardware behavior (e.g., Spectre, Meltdown)
- **IoT Security Flaws** – Weak encryption in smart devices

D. Human & Social Engineering Vulnerabilities

Attackers exploit **human behavior** to gain access.
✅ **Examples:**

- **Phishing** – Tricking users into revealing sensitive information
- **Weak Passwords** – Easy-to-guess credentials
- **Insider Threats** – Employees leaking information

E. Cryptographic Vulnerabilities

Weaknesses in **encryption protocols**.
✅ **Examples:**

- **Weak Encryption (e.g., MD5, SHA-1)** – Easily cracked hashes
- **Poor Key Management** – Hardcoded or exposed cryptographic keys

---

📊 2. How to Assess Vulnerability Severity?

The severity of a vulnerability depends on **exploitability, impact, and affected systems**. The **Common Vulnerability Scoring System (CVSS)** is widely used for assessment.

⚡ CVSS Scoring System

Severity	CVSS Score	Risk Level
🔴 **Critical**	9.0 - 10.0	Immediate threat, actively exploited
🟠 **High**	7.0 - 8.9	Exploitable with serious impact
🟡 **Medium**	4.0 - 6.9	Requires effort to exploit
🔵 **Low**	0.1 - 3.9	Minimal risk, unlikely to be exploited
⚪ **Informational**	0.0	No direct security impact

📌 CVSS Breakdown

A CVSS score is calculated based on **three key factors**:

1️⃣ **Base Score (0-10)**

- **Exploitability** (How easy is it to exploit?)
- **Impact** (Data exposure, system availability, etc.)

2️⃣ **Temporal Score (Adjusts Based on Current Threats)**

- **Is there a public exploit available?**
- **Has the vulnerability been patched?**

3️⃣ **Environmental Score (Adjusts for Specific Systems)**

- **How critical is the affected system?**
- **Can the impact be reduced with security measures?**

✅ **Example:**

- A **publicly available RCE exploit** affecting **a critical server** might score **9.8 (Critical)**.
- An **SQL injection vulnerability** that requires authentication could be **6.5 (Medium)**.

🎯 3. How to Prioritize Fixing Vulnerabilities?

🔺 High Priority (Critical & High)

✅ **Fix Immediately**

- Publicly exploitable vulnerabilities
- RCE, privilege escalation, authentication bypass
- Known active exploitation in the wild

🟡 Medium Priority (Fix in a Timely Manner)

✅ **Fix within a reasonable timeframe**

- SQLi, XSS, and configuration issues
- No known public exploits but still a risk

🔵 Low Priority (Monitor & Plan for Fixes)

✅ **Fix when possible**

- Minimal impact vulnerabilities
- Hard-to-exploit issues

🚀 Final Thoughts

To manage vulnerabilities effectively:
✓ Identify **what type of vulnerability it is**
✓ Use **CVSS scoring** to assess its risk
✓ Prioritize remediation based on **severity and exploitability**
✓ Implement **patches, mitigations, and security best practices**